
This is the **published version** of the bachelor thesis:

Hurtado Ortiz, Antonio; Musach, Ramón, dir. Reserva d'espais en centres educatius. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/238430>

under the terms of the  license

Reserva d'espais en centres educatius

Antonio Hurtado Ortiz

Resum—Aplicació web enfocada a cobrir les necessitats dels centres educatius d'assignar espais a hores lectives, i reservar-los per a quadrar els horaris als espais. Aquest procés d'assignació també ha de permetre consultar els espais disponibles, reservar espais i material de forma dinàmica, i reportar incidències i desperfectes dels mateixos. També ha de tenir en compte l'aforament dels espais a assignar, per a què en una reserva o assignació, aquest no es superi. Per part dels usuaris, l'aplicació ha de comptar amb diferents rols, per a mantenir un funcionament ordenat i correcte, i amb la possibilitat de consultar estadístiques d'ús.

Paraules clau—Aplicació, web, aules, espais, control, gestió, assignació, horaris, material, calendari, reserva, aforament.

Abstract—Application web focused on covering the education centers necessities to allocate classrooms and specific locations with class groups and fit in with the timetable of each centre and cours. The allocation process has to be dinamyc in the way of see disponibility and book the differents locations and material, at the same time that is used to report any incidence or flaw in any of each spaces. It is also important to take in count the capacity that the room or service has, in order to do not overbook it. The application users would have diferent rolls in the way to warranty a correct and organized application running, also they could have the option of check the usage statistics.

Index Terms—Application, web, classrooms, specific locations, control, process, allocation, timetables, material, calendar, book, capacity.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

AQUEST informe exposa el seguiment del projecte de forma continuada, des del moment que es va realitzar la planificació inicial fins a la redacció del mateix.

S'exposaran els objectius actuals, les modificacions que hagin pogut sorgir i els motius de les mateixes, i s'avaluarà el seguiment de la planificació fins a l'actualitat a la línia temporal definida a l'inici.

Tot això permet sintetitzar informació i extreure'n conclusions per saber si el projecte va ben encaminat o si s'ha de revisar algun aspecte i replantejar-lo per poder complir els objectius, tant els propis del projecte com els corresponents al seu procés de desenvolupament: temps, recursos i costos.

Primerament es parlarà dels objectius fixats a l'estat inicial del projecte, i com aquests han anat variant. També s'explicarà l'estat de l'art, apartat en el que s'estudia la situació actual a l'entorn que justifica la proposta del projecte. Seguidament es farà menció a la metodologia escollida, aplicada concretament a aquest projecte de desenvolupament de software. Un cop introduïda la metodologia, s'explicarà la planificació i el seguiment realitzats, i la gestió del projecte. Finalitzada aquesta part, s'explicarà el procés

d'anàlisi i implementació del projecte, procés en el que es desenvolupa l'aplicació, i els resultats i proves realitzades. S'esmentaran les conclusions obtingudes durant tot el projecte, i es tancarà el document amb la bibliografia i altres recursos.

Es recull tot el procés evolutiu del projecte, durant el que es realitza la implementació d'un sistema de gestió d'espais, horaris i reserves per a centres educatius, amb compatibilitat amb dades proporcionades per aplicacions de tercers.

Es pretén incorporar altres funcionalitats que la faran més interessant, com gestió de grups d'usuaris, diferents nivells de permisos, gestió de materials i incidències, notificaciones varies i altres mètodes d'autenticació externes a l'aplicació, propis d'entorns corporatius.

A més, es vol utilitzar una tecnologia de desenvolupament web basada en frameworks d'alt nivell, per a implementar l'aplicació de forma més àgil, modular i robusta.

Per últim, l'aplicació serà accessible des de múltiples plataformes, ja sigui en PC, tauleta o smartphone. Per això, i orientant el desenvolupament a crear altres mètodes d'interactiu, com aplicacions natives, es faran servir tecnologies REST.

2 OBJECTIUS I CANVIS

Durant la fase final del desenvolupament del projecte s'ha avaluat per darrera vegada la viabilitat dels objectius i la seva prioritat.

- E-mail de contacte: antonio.hurtadoo@e-campus.uab.cat
- Menció realitzada: *Tecnologies de la Informació*
- Treball tutoritzat per: Ramon Musach Pi (*Enginyeria de la Informació i de les Comunicacions*)
- Curs 2020/21

Taula 1: OBJECTIUS PRINCIPALS DE L'APLICACIÓ

Objectius	Prioritat
Maquetació web mitjançant una plantilla de tipus dashboard amb Bootstrap, html5 i css3.	Alta
Base de dades MongoDB.	Alta
Rols d'usuari administrador, professor, i alumne.	Alta
L'administrador ha de poder importar actius, usuaris i horaris.	Alta
L'administrador ha de poder donar d'alta actius, usuaris i horaris.	Alta
L'administrador ha de poder crear grups i assignatures.	Alta
L'administrador ha de poder associar horaris i sales, per a assignar espais segons franjes horàries de forma periòdica.	Mitjana
L'aplicació ha de tenir en compte l'aforament màxim dels espais, i el tamany dels grups a assignar.	Baixa
L'administrador i els professors han de poder fer reserves d'espais lliures de forma puntual, independentment dels horaris.	Alta
L'administrador i els professors han de poder eliminar reserves, si en tenen permisos (s'entén que l'administrador en tindrà en qualsevol cas).	Alta
L'administrador, els professors i els alumnes han de poder visualitzar els horaris i les aules que hi tenen assignades, si en tenen permisos.	Mitjana
L'administrador i els professors han de poder visualitzar un calendari amb les sales reservades.	Mitjana
L'administrador i els professors han de poder filtrar sales lliures per franjes horàries.	Mitjana
L'administrador i els professors han de poder visualitzar les franjes reservades per a una sala concreta.	Mitjana

Taula 2: FUNCIONALITATS ADDICIONALS (prioritat baixa)

L'administrador i els professors han de poder reservar materials, inclús independentment dels espais.
Llista de membres dels grups.
Notificacions i alertes a l'aplicació web.
Notificacions i alertes al correu electrònic.
Compatibilitat amb entorns LDAP / Active Directory
L'administrador i els professors han de poder reportar incidències i desperfectes sobre els actius.
L'administrador i els professors han de poder consultar les estadístiques d'ús dels espais i el material.

Taula 3: COMPARATIVA D'EINES

	Plugins en Portal.EDU	SuperSaas	Symplebook.me	Timify
Requisits	Wordpress	-	-	-
Personalització	Creació de recursos i de d'espais de reserva	Determina freqüència, antelació de reserva o cancel·lació i mínim de reserves	Codificació per colors per al persona	Transformació de reunions en reunions virtuals
Visualització del calendari	Sí, amb diferents filtres	Sí, amb diferents filtres	Sí, amb diferents filtres	Sí, amb diferents filtres
Rols d'usuari	Sí	Sí	Sí	Sí
Integració amb altres plataformes	-	Web o pàgina de Facebook amb un iframe	Reserves des de Instagram y Facebook	Reserves des de la intranet, sitio web, Facebook i altres plataformes
Sincronització amb calendari	No	Sí	Sí	Sí
Enviamet de confirmacions	No	Sí	Sí	Sí
Informes i anàlisis	-	Sí	Sí	Sí
Responsive	-	Sí	Sí	Sí
Preu	-	Versió gratuïta i diferents paquets de pagament	Prova gratuïta, després de pagament	Versió gratuïta i diferents paquets de pagament

3 ESTAT DE L'ART

Avui dia és possible trobar eines de gestió per a centres educatius enfocades a múltiples usos. Per exemple, per a gestió d'aules virtuals trobem Moodle, Alexia o Classlife, entre d'altres. Totes elles tenen en comú la gestió de grups i els seus horaris, a banda de la feina que desenvolupen professors i alumnes. Però cap d'elles aporta gestió d'espais. Veiem a la Taula 4 una comparativa d'eines enfocades a resoldre aquesta tasca.

Per últim, trobem altres eines enfocades a la gestió d'inventari, com per exemple GLPI, que mitjançant una sèrie de complements pot assolir totes les competències que es demanen a aquest projecte, però tant la gestió com l'ús de l'eina és poc intuïtiu, i visualment poc atractiu.

Tot això conclou amb la necessitat de desenvolupar una aplicació específicament dissenyada per als objectius que es plantegen al projecte.

4 METODOLOGIA

4.1 Metodologia escollida

Per a aquest projecte de desenvolupament software s'ha decidit optar pels beneficis d'una metodologia de desenvolupament iteratiu i incremental.

Aquesta metodologia aplicada al desenvolupament de software es caracteritza per iniciar el projecte amb una implementació simple, la qual es va complimentant segons avança la planificació, retroalimentant-se amb els requeriments que es van implementant.

El projecte es divideix en blocs que, un cop completats a cada fita, proporcionen una versió funcional amb els requeriments incorporats.

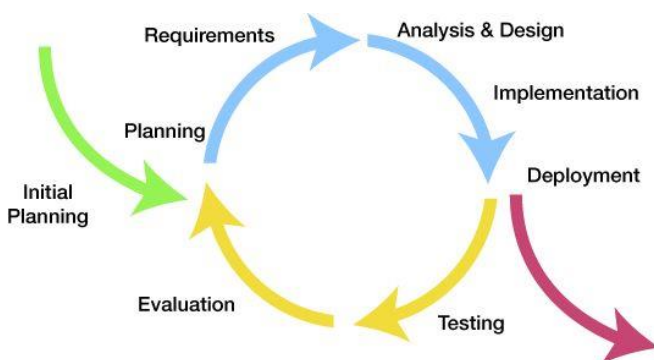
Com a primer pas, és important marcar una prioritat als requisits a implementar, doncs seran presents a primeres

versions del software. Fet així, el projecte s'assegura complir amb els requisits funcionals més importants, encara que s'endarrereixi l'entrega final, o no s'arribi a instaurar tot. Els primers requisits a desenvolupar són els que aporten major valor a l'eina per part del client, els requisits centrals.

Un cop implementats aquests requisits, el software es pot anar complimentant amb altres requisits, que aporten un valor afegit al producte de cara al client.

Veiem un diagrama de les fases del desenvolupament mitjançant el desenvolupament iteratiu i incremental a la figura següent:

Il·lustració 1: AGILE DEVELOPMENT



Com es pot apreciar al diagrama, un cop feta la planificació inicial, es passa a planificar i analitzar els requeriments de cada bloc a desenvolupar.

Un cop implementat, aquest bloc es testreja, documenta i avalua, i es pot començar amb el següent bloc, que repeteix aquest procés. Això permet tenir una versió funcional, encara que incompleta del software, que el client pot anar revisant i donant feedback. D'aquesta manera, en cas de no haver analitzat correctament algun dels requisits funcional, dona una visió dels canvis a realitzar, i permet redirigir el projecte. Per últim, aquesta metodologia dona una visió directa de l'evolució del projecte i, en cas de preveure no finalitzar-lo en la data prevista, facilita el fet de reajustar temps.

4.2 Aplicació de la metodologia al projecte

La plataforma web es desenvolupa en Node.js [1] i Vue [2], principalment. S'estructura com una eina client-servidor, sota peticions de tipus REST [3] i respostes mitjançant estructures JSON [4].

Es desenvolupa amb eines locals en un entorn Windows, amb servidors Node i Vue executats localment. Un cop es completa una nova funcionalitat, es procedeix a actualitzar els repositoris a Github [5], eina escollida per a la gestió de versions i repositoris. Trobem un repositori per al front-end i un altre per al back-end, ja que es tracta de dos parts ben diferenciades.

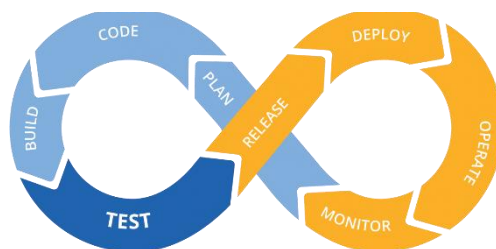
Actualitzats els repositoris, es procedeix a actualitzar els arxius al servidor Heroku [6]. Aquest integra una funcionalitat "Deploy" que, tenint vinculat el repositori de la banda del servidor (que també inclou el front-end web a servir), i correctament configurat l'arxiu *package.json* amb els mòduls i els scripts corresponents al projecte, és capaç

de resoldre les dependències necessàries, importar els arxius des de Github i posar en marxa el servidor de nou, oferint la nova funcionalitat actualitzada envers als usuaris.

Aquest tipus d'implementació s'anomena CI/ CD (Continuous Integration / Continuous Delivery) [7], el que facilita implementar l'aplicació i entregar funcionalitats de forma contínua i gairebé transparent envers al client final.

És especialment eficaç conjuntament amb una metodologia com la escollida per a aquest projecte, de tipus iterativa i incremental, doncs amb aquest desenvolupament es duu a la pràctica la metodologia, implementant de forma contínua funcionalitats envers al client, de la forma més transparent possible.

Il·lustració 2: CI/CD



5 PLANIFICACIÓ I SEGUIMENT

Des d'un inici es van plantejar uns terminis per dur a terme el projecte i poder-lo tenir finalitzat de cara a la data d'entrega. Tot i així és normal que, durant un projecte, a causa de diversos factors es puguin veure alterades les dates reals de terminis de cada procés. Portar un control ajuda a redirigir o reorganitzar el projecte per tal de poder complir els objectius marcats. Veiem el nivell d'acompliment de les dates durant els següents apartats:

5.1 Nivell de seguiment

Degut als esdeveniments plantejats durant l'evolució del projecte, s'ha optat per prescindir de la majoria de funcionalitats addicionals, el que presenta un cert risc envers al nivell de satisfacció dels clients finals del producte. No obstant, la decisió ha contribuït a esdevenir la entrega del producte final dintre de les dates planificades.

5.2 Ajustaments efectuats

Durant la primera fase del desenvolupament del projecte es van trobar dificultats relacionades amb l'adaptació de la plantilla escollida, doncs al estar confeccionada mitjançant Vuetify [8] i no tenir formació al respecte, assolir els coneixements necessaris va suposar un retard.

A banda, l'ús de NodeJS com a back-end i Vue com a front-end va presentar el paradigma de necessitar un intercanvi d'informació segur per a realitzar les validacions necessàries respecte als usuaris i els seus rols. Es va solventar amb l'ús de tokens mitjançant JWT [9]. Tot això també va requerir de formació prèvia, i impossibilitava la integració amb Active Directory [10] o LDAP [11]. Es va decidir prescindir d'aquesta funcionalitat addicional.

A més, fent una cerca sobre la capacitat de realitzar enviaments de correu electrònic des de NodeJS, no es va trobar una integració ràpida amb aquest servei, amb la qual cosa es va decidir suspendre i posteriorment suprimir aquesta funcionalitat.

Al ser funcionalitats amb baix impacte sobre les necessitats dels usuaris finals, no s'espera un alt increment del risc sobre el projecte, i es flexibilitza el temps a emprar per a implementar les funcionalitats principals que portin resultats.

A més, durant la tercera fase del desenvolupament del projecte es van trobar dificultats relacionades amb la posada en marxa del servidor a Heroku, en el que s'executa el back-end, i el servidor a MongoDB Atlas [12], en el que s'emmagatzema la base de dades.

Concretament, el mòdul Babel, utilitzat per a generar codi javascript universal, entenedor des de qualsevol navegador des de qualsevol font, va fallar. El motiu va ser la seva utilització com a mòdul de desenvolupament, definint-lo com a dependència de l'entorn *dev*. Es va resoldre la incidència situant la dependència a l'entorn de producció, i actualitzant el projecte al servidor.

La implementació de la connexió entre el back-end i la base de dades al núvol amb MongoDB Atlas va suposar també una dificultat afegida, en aquest cas resolta de forma ràpida. Analitzant la documentació proveïda per la mateixa plataforma, resulta relativament senzill realitzar la connexió entre el projecte back-end amb Node.js (a través del mòdul Mongoose [13]), i la base de dades en Atlas.

Durant la última fase del desenvolupament del projecte s'han trobat altres dificultats, entre elles destacables les següents:

La necessitat de redistribuir el menú de navegació, doncs a la part superior no es feia pràctic d'utilitzar. S'utilitza una de les plantilles pròpies de Vuetify, amb el menú lateral a l'esquerra i el contingut de la web al centre, a un gran contenidor.

El fet de treballar amb una base de dades no relacional com MongoDB dificulta l'establiment de relacions entre taules i, en conseqüència, l'extracció de dades de forma combinada i la propagació de canvis al modificar o eliminar registres dels que altres dades en depenguin.

Es soluciona aquest problema establint sobre Mongoose models específics per a aquests casos al backend de l'aplicació web.

Per concloure aquest apartat, s'han retirat la majoria de característiques de prioritat baixa degut a dificultats derivades de la implementació de la funcionalitat de cerca de sales disponibles.

6 GESTIÓ DEL PROJECTE

6.1 Terminis

L'avaluació conclou sense observar canvis envers a la planificació inicial d'entrega, tot i que per assolir-ho s'han descartat la majoria de funcionalitats de baix impacte.

6.2 Recursos

Els recursos necessaris per a dur a terme el projecte són fonamentalment els recursos humans. Es compta amb una

persona, que durà a terme tots els rols necessaris.

A banda, com a material tècnic, es comptarà amb una estació de treball amb MS Windows 10, accés a Internet, i els perifèrics adequats per a dur a terme les tasques. Com a software necessari, es compta amb una suite d'ofimàtica MS Office, amb llicència de MS Project, per a la planificació del projecte, l'elaboració de fulls de càlcul, la redacció d'informes, i la presentació final. També serà necessari comptar amb l'entorn de desenvolupament integrat adequat, el qual passa de ser Webstorm, de JetBrains, a VisualStudio Code. Com a repositori s'ha escollit utilitzar Github, i com a servidors de desenvolupament un servidor local, i un servidor remot gratuït a Heroku. S'afegeix l'ús de MongoDB en la seva versió Cloud, de forma gratuïta.

Durant aquesta part del desenvolupament també s'ha requerit de l'ús de les següents eines: GitKraken, per a la gestió dels repositoris a GitHub. PowerShell, per a l'execució de certes comandes en local, i per a l'accés a la terminal de Heroku. I per últim, MongoDBCompass, com a client gràfic de MongoDB.

Per últim, la referència trobada al curs de MEVN de Bluuweb [14]. Ha estat un recurs i referència invaluables per a dur a terme el projecte.

6.3 Cost

Les hores treballades les podem veure reflectides a la planificació del projecte: 117 dies, que suposen 315 hores de treball. Assumint que l'hora treballada es remuneri a 35€, el cost dels recursos humans del projecte ascendeix als 11.025€. El projecte ja compta amb les llicències d'ús de tot el software que farà falta. Per tant, comporta un cost de 0€. Per tant, el cost total del projecte se situa en 11.025€. No s'observen canvis des de l'informe inicial.

6.4 Riscos o imprevistos

Durant el desenvolupament del projecte s'han trobat una sèrie de dificultats d'implementació, que han obligat a deixar de banda una part de les funcionalitats plantejades a l'inici. Tot i així, aquestes característiques descartades han estat de baix impacte, i no s'espera un impacte rellevant en l'acceptació del producte. No obstant, son funcionalitats desitjables per a arribar a un nombre màxim de clients, i comporta cert risc en l'acceptació i comercialització del producte, si escau.

6.5 Lliurables

Fins aquest informe final s'han realitzat tres lliurables, és a dir, s'han fet les tres primeres iteracions del sistema.

El primer lliurable correspon a la planificació del projecte, l'avaluació de riscos i costos, i la presa de decisions respecte a les tecnologies a utilitzar i la metodologia.

El segon lliurable correspon a la re-estructuració del projecte si escau, i la avaluació general dels costos, riscos i compliment de dates.

El tercer lliurable correspon a la revisió de la Iteració 2 conjuntament amb les iteracions anteriors. Es compara la planificació modificada a l'avaluació anterior amb l'acompliment de la planificació durant aquesta darrera part del desenvolupament.

L'informe final, és a dir, aquest mateix lliurable,

correspon a la síntesi de tota la evolució del projecte. Es plasma l'evolució durant la última fase del desenvolupament del projecte, i s'hi plasma informació rellevant envers al desenvolupament, proves, resultats i conclusions del mateix.

7 ANÀLISIS

S'avaluen les funcionalitats requerides i s'elabora un diagrama de casos, molt útil per a visualitzar què funcions s'han d'implementar, i elaborar-ne proves un cop implementades.

Primerament, es veu la necessitat d'incorporar 3 rols diferenciats segons privilegis:

Un rol USER, limitat, amb privilegis suficients per a visualitzar les reserves a les que hi té accés, però sense cap poder d'edició. Un rol PROFESOR, amb capacitat per a crear i associar contingut de qualsevol tipus, però amb accés limitat al seu propi contingut. Per exemple, un professor pot crear, modificar o eliminar grups, assignatures, o reserves, sempre que les hagi creat ell mateix. Per últim, un rol ADMIN, amb capacitat per a visualitzar, crear, modificar o eliminar contingut, ja sigui propi o d'un altre usuari, sense cap mena de límit.

L'aplicació utilitza un sistema d'autenticació mitjançant un token JWT. Un cop l'usuari s'autentica a la pàgina de login, el token es crea al back-end i es retorna al front-end, a on s'emmagatzema localment. Tota vista a la que l'usuari intenti accedir, i tota petició que el front-end realitzi al back-end, validarà el token i en consultarà el rol. Si el token no és vàlid, l'aplicació descartarà el token, i en forçarà la redirecció al formulari de login.

Il·lustració 3: IMPLEMENTACIÓ JWT A NODE.JS

```

middlewares > JS autenticacion.js > verificarAuth > jwt.verify('secret') callback
1  const jwt = require('jsonwebtoken');
2
3  const verificarAuth = (req, res, next) => {
4
5      const token = req.get('token');
6
7      jwt.verify(token, 'secret', (err, decoded) => {
8          if(err){
9              return res.status(400).json({
10                 mensaje: 'Usuario no válido',
11                 err
12             })
13         }
14
15         req.usuario = decoded.data;
16
17         next();
18     });
19 }
20
21 }
22
23 const verificarAdministrador = (req, res, next) => {
24     const role = req.usuario.role
25
26     if(role === 'ADMIN'){
27         next();
28     } else {
29         return res.status(401).json({
30             mensaje: 'Usuario no válido'
31         })
32     }
33 }
34

```

Aquest tipus d'aplicació, seguint la línia d'una aplicació REST amb front-end i back-end separats, presenta

flexibilitat envers a comptar amb diferents front-ends accedint al back-end, sense implementar noves funcionalitats. Per exemple, tenint una aplicació Android accedint de forma concurrent amb el front-end desenvolupat a aquest projecte, qualsevol part del back-end inclosa l'autenticació mitjançant token seria vàlida i compatible.

La principal funcionalitat a implementar és la reserva d'espais envers a una franja horària. És necessari implementar els mètodes i les connexions a la base de dades necessàries per a emmagatzemar els períodes, les franjes horàries, els grups, els espais, els usuaris, i les reserves.

Un cop es tenen mètodes per a treballar amb les dades al back-end, es procedeix a implementar les funcionalitats que demanen aquestes dades des del front-end, i les mostren a l'usuari que hi accedeix.

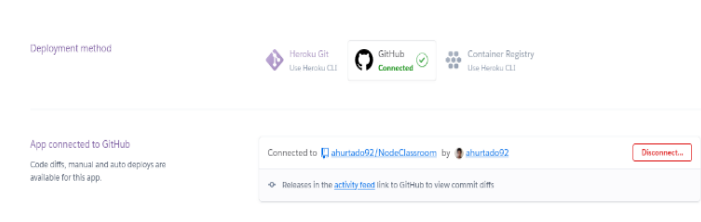
Altres funcionalitats implementades son la inclusió d'un apartat de notes, privades per a cada usuari, la gestió de material reservable, i d'incidències sobre els espais i el material.

8 IMPLEMENTACIÓ

8.1 Necessitats tècniques

Primerament ha estat necessari configurar les aplicacions i espai de treball. S'instala a una màquina amb Windows 10 els navegadors Mozilla Firefox i Google Chrome, l'editor de text Visual Studio Code, la CLI de Heroku i de Git, l'aplicació Gitkraken [15] i l'extensió Vue Dev Tools per als navegadors usats. Un cop instal·lades les eines necessàries, es procedeix a configurar les CLI amb els comptes d'usuari de Heroku i Github, i l'accés des de PowerShell a aquestes CLI. També s'instala i configura MongoDB Compass com a client de la base de dades MongoDB Atlas, i Insomnia com a client de peticions https per a iniciar el desenvolupament del back-end. S'inicien els projectes de front-end i back-end a Github amb un primer commit i es procedeix a vincular el repositori de back-end a Heroku.

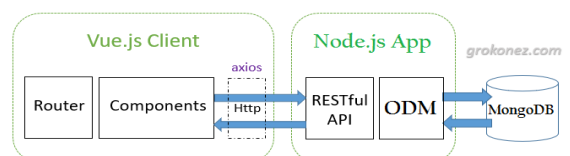
Il·lustració 4: GITHUB VINCULAT A HEROKU



8.2 Desenvolupament

Un dels principals avantatges dels llenguatges d'alt nivell és la facilitat que brinden per a reaprofitar codi.

Il·lustració 5: COMUNICACIÓ API



En el cas de Node.js i Vue la modularitat representa en gran mesura aquesta capacitat. S'utilitzen mòduls per a implementar capacitats i funcionalitats desenvolupades per tercers. Aquests mòduls (node modules) es defineixen a l'arxiu package.json, a l'arrel del projecte, tant a Node.js com a Vue.

El procés de desplegament és el següent:

1. Es crea la distribució del projecte front-end amb Vue, i es copia el resultat generat a la carpeta "public" del back-end amb Node.js.
2. Es publica el projecte back-end al repositori de GitHub. Mai es publica el contingut dels "Node Modules", ja que al fer el *deploy* de l'aplicació, el servidor instal·larà les dependències necessàries especificades al package.json.
3. S'actualitza l'aplicació web des del panel de Heroku. Aquest procés para l'aplicació, i importa el nou codi actualitzat al servidor web, instal·lant-ne les dependències necessàries, comprimint el codi resultant, i tornant a aixecar els serveis necessaris per al funcionament del projecte.

Un cop l'usuari accedeix al servidor web, aquest li retorna l'aplicació front-end, que queda carregada a la banda de client, al navegador web. Aquesta s'encarregarà de realitzar les peticions REST corresponents (GET, POST, PUT i DELETE) cap a les rutes corresponents. Un cop el back-end rep la petició i els paràmetres necessaris, la processa mitjançant els models declarats a Mongoose, en recull la informació de MongoDB, i retorna al front-end la resposta en format d'arxiu JSON. Aleshores, el front-end interpreta la resposta i la tracta per a mostrar-la a l'usuari en funció de la vista, el rol i els permisos.

8.2.1 Back-End

Node utilitza Mongoose com a gestor de BD. Aquest mòdul s'encarrega de tractar amb la base de dades mitjançant models. Els models són un objecte de tipus Schema, amb mètodes propis per a facilitar el tractament de dades entre la BD i Node.js.

Il·lustració 6: MODEL GROUP

```
models > $ group.js > ...
1 import mongoose from 'mongoose';
2 const Schema = mongoose.Schema;
3
4 const groupSchema = new Schema({
5   name: {type: String, required: [true, 'Nombre obligatori.'],
6   description: String,
7   creatorId: String,
8   members: [],
9   date: {type: Date, default: Date.now}
10 });
11 // Convertir a un modelo
12 const Group = mongoose.model('Group', groupSchema);
13
14 export default Group;
```

A banda de Mongoose, s'han utilitzat altres mòduls o middlewares [16] per al desenvolupament del back-end: Actualment, Node.js no soporta ECMAScript 6 [17] de forma nativa. S'utilitza Babel [18] per a generar una traducció de codi JS, el que permet executar al navegador codi ES6 amb totes les seves funcionalitats.

Express és el middleware que permet a Node.js realitzar

direccionaments mitjançant la declaració de rutes al Router [19]. Quan el Front-End realitza una petició a una ruta declarada al Back-End, aquest la interpreta, i n'executa una resposta. Juntament amb Express, Cors [20] facilita l'accés a recursos de Node.js des del Front-End mitjançant funcions REST.

Bcrypt [21] és el middleware encarregat d'encriptar a la base de dades la contrasenya dels usuaris.

JSON Web Token realitza totes les gestions necessàries per a crear un token de sessió per a l'usuari, validar-lo i proporcionar accés a les rutes.

Moment [22] proporciona mètodes per a tractar amb formats horaris.

Morgan [23] és un middleware de desenvolupament, el que vol dir que només està present a l'entorn dev (local). S'encarrega de generar una sortida per la consola del servidor back-end en resposta a les peticions que li arriben. Aquesta informació acilta el desenvolupament.

Il·lustració 7: MORGAN

```
> nodemon app.js --exec babel-node
[nodemon] 2.0.6
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `babel-node app.js`
```

A banda dels middlewares, a l'entorn de desenvolupament s'ha emprat Nodemon [24], una eina que reinicia el servidor automàticament cada vegada que es realitza un canvi als arxius del Back-End. Això evita haver d'estar reiniciant el servidor manualment cada cop que es modifica el codi.

Il·lustració 8: NODEMON

```
Example app listening on port: 3000
DB connected
OPTIONS /api/notas 204 0 - 3.123 ms
GET /api/notas 200 202 - 83.553 ms
OPTIONS /api/groups 204 0 - 1.295 ms
OPTIONS /api/bookings 204 0 - 0.269 ms
OPTIONS /api/rooms 204 0 - 0.265 ms
GET /api/rooms 304 - - 66.884 ms
GET /api/groups 304 - - 82.454 ms
GET /api/bookings 304 - - 374.044 ms
[]
```

Els control·lers són els encarregats de la part lògica del Back-End. Un cop arriba una petició a una ruta declarada, aquesta petició executa una funció al controlador corresponent. Aquest valida el token de l'usuari que li envia el front-end, en recull les variables de la petició si escau, i sol·licita a l'objecte mongoose (el model) el tractament amb la base de dades necessari. Quan la interacció amb la base de dades finalitza, el controlador en recull una resposta, i tant si és satisfactòria com no, s'envia al front-end per a servir una resposta a l'usuari.

Veiem com a exemple la mecànica del controlador Users en resposta a la ruta declarada per a sol·licitar un usuari envers el seu ID:

Il·lustració 9: USUARI PER ID

```

41 // Get con parámetros
42 router.get('/usuario/:id', async(req, res) => {
43
44   const _id = req.params.id;
45
46   try {
47     const userDb = await User.findOne({_id});
48     res.json(userDb);
49   } catch (error) {
50     return res.status(400).json({
51       mensaje: 'Ocurrió un error',
52       error
53     });
54   }
55 });

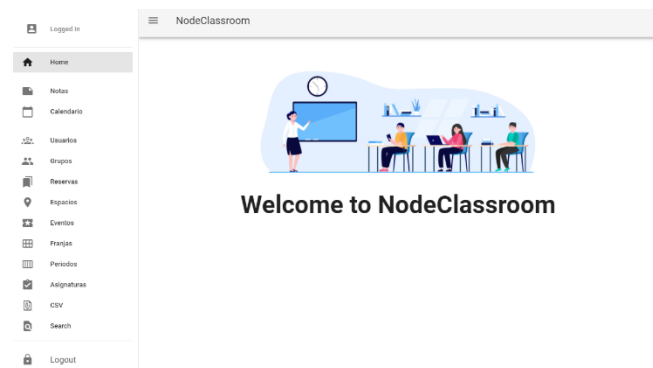
```

A la il·lustració anterior veiem com es recull el ID com a paràmetre a la ruta de tipus get. A continuació es declara una promesa, que s'acompleix quan es consulta al model per l'usuari amb aquella ID, i es retorna al Front-End la resposta en format JSON. Si aquesta acció no es pot dur a terme, es captura l'error generat, que es retorna amb un missatge i codi d'error 400 al client.

8.2.2 Front-End

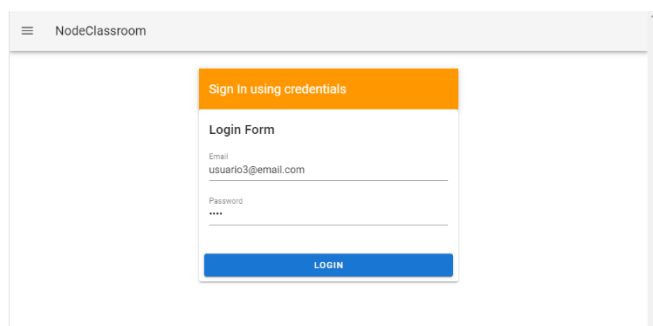
El Front-End es serveix des del directori *public* del servidor quan es sol·licita la web a través d'un navegador. Un cop carregat, aquest valida el token de sessió que, de estar present i ser vàlid, proporciona accés a l'aplicació.

Il·lustració 10: VISTA PROFESSOR



En cas de no existir aquest token, només s'hi tindrà accés al Home i al formulari d'inici de sessió. Un cop l'usuari introdueix les dades de login, el front-end les recull en un objecte de tipus json, que envia al back-end. Aquest les recull i consulta a la base de dades. Si les dades són correctes, el back-end genera un token d'accés, que fa arribar al front-end mitjançant una resposta en format json. En cas de no ser correctes, es genera una

Il·lustració 11: LOGIN



resposta d'error, i no es permet l'inici de sessió.

Si el token d'usuari és present, i l'usuari és de tipus Alumne, només tindrà accés a l'apartat de Home, Calendari i Logout. En canvi, un usuari de tipus Professor o Administrador tindrà accés al menú de forma completa. L'usuari Professor tindrà accés al contingut del qual sigui propietari, mentre que l'Administrador tindrà accés complet a tot el contingut, en sigui o no el propietari. A partir d'aquí, l'usuari pot navegar i interactuar amb l'aplicació.

Vue és un framework per a treballar amb Javascript orientat a Front-End, generant una aplicació en una única vista que renderitza tot el contingut. A aquest tipus d'aplicació se'ls denomina Single-Page Application o SPA.

De la mateixa manera que Node.js, Vue pot treballar amb mòduls de tercers, i Components [25] externs. Veiem a continuació els que s'han utilitzat en aquest desenvolupament.

Axios [26] s'encarrega de realitzar les peticions cap al Back-End, i en gestiona la resposta en forma de JSON. Al realitzar una petició a través d'una ruta, Axios pot encapsular tot el necessari al head o al body de la comunicació. Un cop s'obre el canal, n'espera una resposta, o bé en forma de codi, o bé en forma de JSON. S'executa en forma de promesa, de forma asíncrona, pel que pot carregar la resposta a posteriori de carregar la pàgina, quan aquesta s'executi i el client rebi dades des del servidor.

Igual que a Node.js, Moment s'utilitza per a gestió d'objectes de tipus data i hora.

Com ja s'ha esmentat abans, Vue treballa com a SPA. Incorpora un mòdul per a realitzar peticions sobre el mateix Front-End, Vue-Router, que en aquest cas es configura amb el seu mode History per a simular diverses pàgines sobre la SPA.

En aquesta aplicació s'emmagatzemen diversos mètodes i variables que han d'estar disponibles de forma transversal a tota l'aplicació, com tot el referent a l'usuari loguejat. Store [27] és un mòdul que proporciona aquesta funcionalitat. JWT-Decode [28] proporciona mètodes per a tractar amb el token de sessió que retorna el Back-End. És gràcies a aquest complement que el Front-End pot validar l'usuari a cada ruta.

Vuetify vitamina les propietats i l'estil de Vue, afegint components que de forma transparent agilitzen la implementació d'una distribució i estil concrets de forma similar a Bootstrap [29] o MaterialDesign, tals com la formació del menú lateral, o l'estil aplicat als formularis i taules.

A més, gràcies a aquest complement, es poden incorporar altres components desenvolupats per tercers utilitzant també Vuetify, tals com V-Calendar [30], DatePicker [31], TimePicker [32] o ColorPicker [33].

Vuex proporciona funcionalitats extra a Vue, tals com Getters, Mutations, States i Actions. Està basat en Props [34], propietat nativa a Vue, i facilita la comunicació entre components i el reaprofitament del codi.

Vue-CSV-Import [35] és el component extern que facilita el procés de pujada del fitxer CSV amb el que s'incorporen les reserves automatitzades a l'aplicació. Costa d'un formulari per a seleccionar els paràmetres i l'arxiu, i de tota la mecànica per a carregar-lo a una estructura de

dades, que posteriorment s'envia al Back-End per a validar i incorporar a la base de dades.

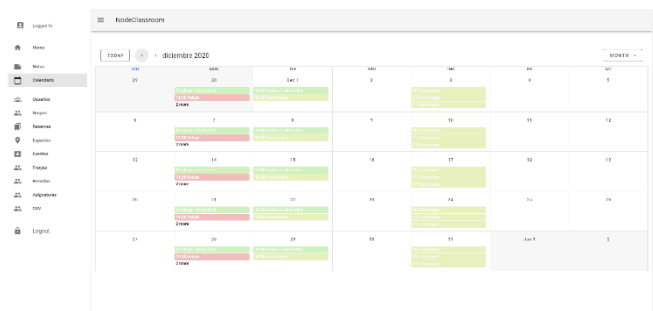
De la mateixa manera que Node.js utilitza Babel per a generar codi ECMAScript 6, Vue utilitza Core-JS [36].

9 RESULTATS

9.1 Resultats

Com a resultat del desenvolupament del projecte trobem una eina amb característiques responsive donades per la maquetació amb Vuetify.

Il·lustració 12: CALENDARI



Al separar el projecte en Front-End i Back-End podem observar principalment dues avantatges: s'optimitzen els temps de càrrega i consums de xarxa, al carregar una única vegada l'entorn, i processant part de la informació a la banda del client, i es facilita la implementació d'altres entorns d'usuari aprofitant les mecàniques i mètodes ja implementats al Back-End.

S'aconsegueix una aplicació capaç d'interpretar i treballar amb l'entrada de dades des d'un fitxer CSV extern donat per una altra aplicació de gestió d'horaris i franjes, que a més facilita la feina al professorat i a l'alumnat a l'hora de consultar reserves al calendari, realitzar-ne de noves i cercar llocs per a activitats puntuals.

9.2 Proves

S'han realitzat diferents proves, tant funcionals [37] com d'usuari [38], donant aquestes un resultat correcte dels comportaments esperats de l'aplicació, i una acceptació força positiva per part de l'equip de professors de l'escola escollida.

Concretament, es va escollir a l'actual cap d'estudis, i a una professora de cicles formatius. Al no comptar amb una eina de planificació d'horaris, se'ls hi va proporcionar un fitxer CSV de prova per a realitzar els tests. Aquests, de forma intuïtiva van entrar a l'aplicació amb les credencials proporcionades, i van importar el fitxer de forma correcta.

A banda, van consultar el calendari tant en PC com en dispositius mòbils, accedint a la vista mensual, setmanal i de dia, i creant, modificant i eliminant tant períodes lectius, franges horàries, grups, usuaris i reserves.

10 CONCLUSIONS

Un cop finalitzat el projecte, podem veure que s'ha aconseguit satisfer les necessitats plantejades a l'inici pel client.

Encara que s'ha limitat la funcionalitat final envers al plantejament inicial, s'ha aconseguit implementar totes les funcionalitats d'alt i mig impacte de forma correcta, i s'ha donat com a resultat una aplicació funcional.

Compta amb modularitat per a implementar funcionalitats noves, és responsive i funcional a qualsevol navegador actual, i té un moderat consum d'ample de banda.

A més ha presentat una acceptació força bona al col·lectiu que l'ha testejat, i el seu ús ha estat força intuïtiu, sense pràcticament necessitar aclariments.

Tot i així, no s'han dut a terme totes les funcionalitats plantejades, cosa que hagués estat força interessant, doncs la implementació d'altres mètodes d'autenticació, la gestió de material i incidències, les estadístiques de reserves, i les notificacions envers al client obririen la porta al mercat per altres fins, no només la gestió d'horaris.

També cal esmentar que seria interessant implementar l'exportació dels events en format iCal, per a incorporar-los a altres calendaris offline, i la sincronització de forma activa amb altres serveis de calendari com Google Calendar o Outlook.

Per últim, cabria proposar, a banda de les funcionalitats abans esmentades, la implementació d'una aplicació nativa per a dispositius mòbils, tant Android com Apple, en format de Smartphone i Tablet, doncs avui dia són eines gairebé imprescindibles a l'aula, i s'integren més amb els dispositius mòbils que les aplicacions web.

Tot i així, el desenvolupament s'ha dut a terme de forma correcta, respectant en grau les iteracions i entregues planificades.

11 BIBLIOGRAFIA

- [1] Node.js, «Node.js», *Node.js*. [Online]. Disponible en: <https://nodejs.org/en/>. [Accedido: 07-feb-2021]
- [2] «What is Vuex? | Vuex». [Online]. Disponible en: <https://vuex.vuejs.org/#what-is-a-state-management-pattern>. [Accedido: 07-feb-2021]
- [3] «¿Qué es REST? Conoce su potencia», *OpenWebinars.net*, 17-may-2018. [Online]. Disponible en: <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>. [Accedido: 07-feb-2021]
- [4] «JSON». [Online]. Disponible en: <https://www.json.org/json-en.html>. [Accedido: 07-feb-2021]
- [5] «GitHub: Where the world builds software», *GitHub*. [Online]. Disponible en: <https://github.com/>. [Accedido: 07-feb-2021]
- [6] «Cloud Application Platform | Heroku». [Online]. Disponible en: <https://www.heroku.com/>. [Accedido: 07-feb-2021]
- [7] Nicolasbisso, «Introducción a CI/CD con github actions», *Medium*, 01-oct-2020. [Online]. Disponible en: <https://medium.com/@nicolasbisso7.1/introducci%C3%B3n-a-ci-cd-con-github-actions-a87902a5b52>. [Accedido: 07-feb-2021]
- [8] «Vuetify — A Material Design Framework for Vue.js», *Vuetify*. [Online]. Disponible en: <https://vuetifyjs.com/>. [Accedido: 07-feb-2021]

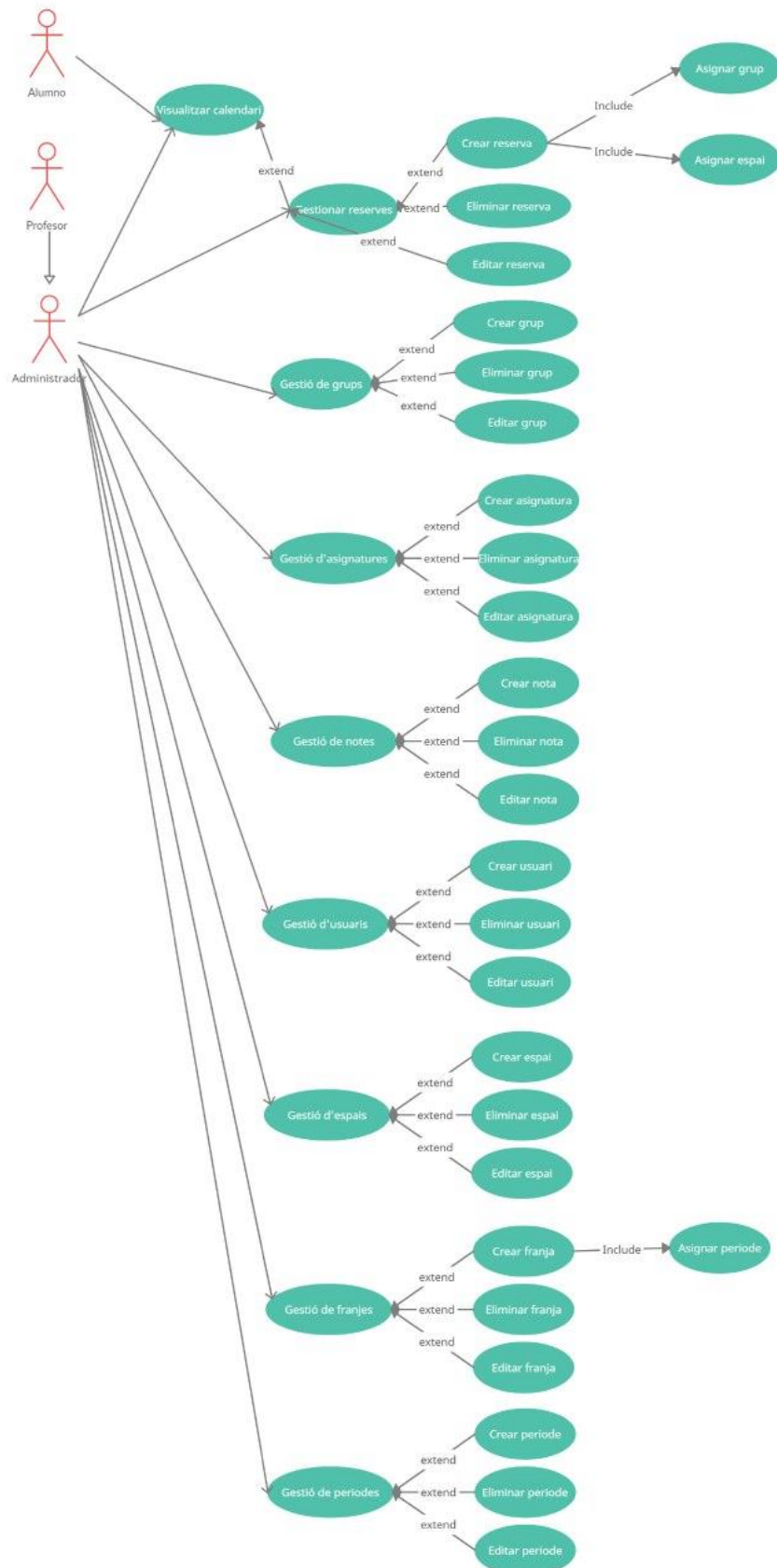
- [9] auth0.com, «JWT.IO». [Online]. Disponible en: <http://jwt.io/>. [Accedido: 07-feb-2021]
- [10] «Azure Active Directory | Microsoft Azure». [Online]. Disponible en: <https://azure.microsoft.com/es-es/services/active-directory/>. [Accedido: 07-feb-2021]
- [11] «PHP: LDAP - Manual». [Online]. Disponible en: <https://www.php.net/manual/es/book.ldap.php>. [Accedido: 07-feb-2021]
- [12] «Managed MongoDB Hosting | Database-as-a-Service», *MongoDB*. [Online]. Disponible en: <https://www.mongodb.com/cloud/atlas>. [Accedido: 07-feb-2021]
- [13] «Mongoose v5.11.15: Models». [Online]. Disponible en: <https://mongoosejs.com/docs/models.html>. [Accedido: 07-feb-2021]
- [14] «Introducción a MEVN | Vue.js, Node.js, Express & MongoDB [MEVN]». [Online]. Disponible en: <https://bluuweb.github.io/mevn/>. [Accedido: 07-feb-2021]
- [15] «Free Git GUI for Windows, Mac, Linux | GitKraken», *GitKraken.com*. [Online]. Disponible en: <https://www.gitkraken.com/>. [Accedido: 07-feb-2021]
- [16] «Qué es middleware: definición y ejemplos | Microsoft Azure». [Online]. Disponible en: <https://azure.microsoft.com/es-es/overview/what-is-middleware/>. [Accedido: 06-feb-2021]
- [17] «JavaScript ES6». [Online]. Disponible en: https://www.w3schools.com/js/js_es6.asp. [Accedido: 06-feb-2021]
- [18] «Babel · The compiler for next generation JavaScript». [Online]. Disponible en: <https://babeljs.io/>. [Accedido: 06-feb-2021]
- [19] «router», *npm*. [Online]. Disponible en: <https://www.npmjs.com/package/router>. [Accedido: 06-feb-2021]
- [20] «Express cors middleware». [Online]. Disponible en: <https://expressjs.com/en/resources/middleware/cors.html#configuration-options>. [Accedido: 06-feb-2021]
- [21] «bcrypt», *npm*. [Online]. Disponible en: <https://www.npmjs.com/package/bcrypt>. [Accedido: 06-feb-2021]
- [22] «Moment.js | Home». [Online]. Disponible en: <https://momentjs.com/>. [Accedido: 06-feb-2021]
- [23] «Express morgan middleware». [Online]. Disponible en: <http://expressjs.com/en/resources/middleware/morgan.html>. [Accedido: 06-feb-2021]
- [24] «nodemon», *npm*. [Online]. Disponible en: <https://www.npmjs.com/package/nodemon>. [Accedido: 06-feb-2021]
- [25] «Components Basics — Vue.js». [Online]. Disponible en: <https://vuejs.org/v2/guide/components.html>. [Accedido: 07-feb-2021]
- [26] «Axios», *Axios*. [Online]. Disponible en: <https://www.axios.com/>. [Accedido: 07-feb-2021]
- [27] «The open-source frontend for any eCommerce | Vue Storefront». [Online]. Disponible en: <https://www.vuestorefront.io/>. [Accedido: 07-feb-2021]
- [28] «vue-jwt-decode», *npm*. [Online]. Disponible en: <https://www.npmjs.com/package/vue-jwt-decode>. [Accedido: 07-feb-2021]
- [29] «BootstrapVue», *BootstrapVue*. [Online]. Disponible en: <https://bootstrap-vue.org>. [Accedido: 07-feb-2021]
- [30] «Introduction | V-Calendar». [Online]. Disponible en: <https://vcalendar.io/>. [Accedido: 07-feb-2021]
- [31] «vuejs-datepicker», *npm*. [Online]. Disponible en: <https://www.npmjs.com/package/vuejs-datepicker>. [Accedido: 07-feb-2021]
- [32] «vue2-timepicker», *npm*. [Online]. Disponible en: <https://www.npmjs.com/package/vue2-timepicker>. [Accedido: 07-feb-2021]
- [33] «Color picker component», *Vuetify*. [Online]. Disponible en: <https://vuetifyjs.com/en/components/color-pickers/>. [Accedido: 07-feb-2021]
- [34] «Propiedades — Vue.js». [Online]. Disponible en: <https://es.vuejs.org/v2/guide/components-props.html>. [Accedido: 07-feb-2021]
- [35] «Vue.js component to handle CSV uploads with field mapping», *Vue.js Examples*, 20-dic-2018. [Online]. Disponible en: <https://vuejsexamples.com/vue-js-component-to-handle-csv-uploads-with-field-mapping/>. [Accedido: 07-feb-2021]
- [36] «core-js», *npm*. [Online]. Disponible en: <https://www.npmjs.com/package/core-js>. [Accedido: 07-feb-2021]
- [37] «Pruebas funcionales / No funcionales: ¿Qué son y para qué sirven?», *Tester House*, 26-mar-2019. [Online]. Disponible en: <https://testerhouse.com/teoria-testing/pruebas-funcionales/>. [Accedido: 07-feb-2021]
- [38] «What is Usability Testing?», *The Interaction Design Foundation*. [Online]. Disponible en: <https://www.interaction-design.org/literature/topics/usability-testing>. [Accedido: 07-feb-2021]

12 INDEX DE FIGURES

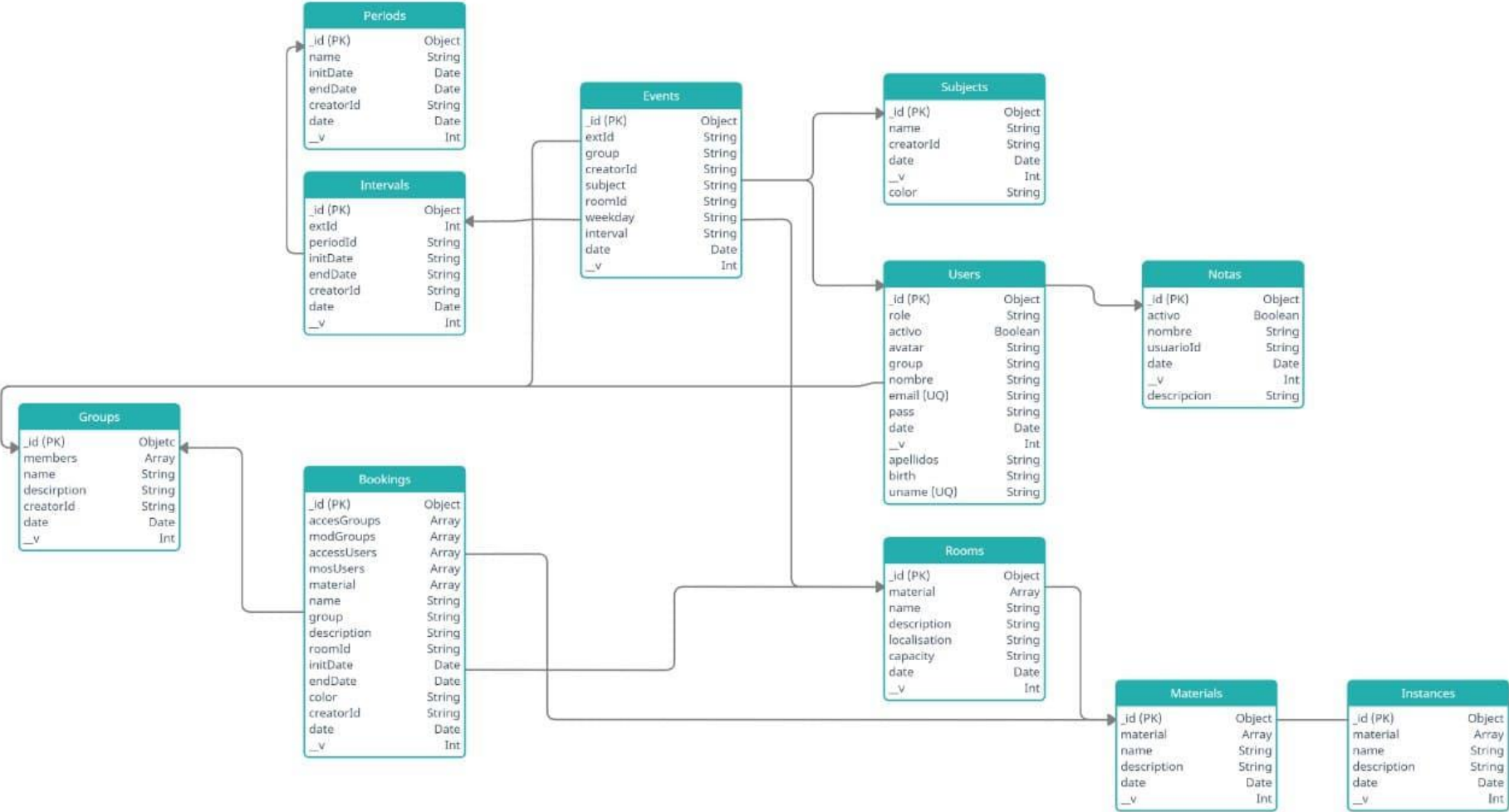
Il·lustració 1: AGILE DEVELOPMENT	3
Il·lustració 2: CI/CD	3
Il·lustració 3: IMPLEMENTACIÓ JWT A NODE.JS	5
Il·lustració 4: GITHUB VINCULAT A HEROKU	5
Il·lustració 5: COMUNICACIÓ API	5
Il·lustració 6: MODEL GROUP	6
Il·lustració 7: MORGAN	6
Il·lustració 8: NODEMON	6
Il·lustració 9: USUARI PER ID	7
Il·lustració 10: VISTA PROFESSOR	7
Il·lustració 11: LOGIN	7
Il·lustració 12: CALENDARI	8
Il·lustració 13: DIAGRAMA DE CASOS	10
Il·lustració 14: DIAGRAMA DE BASE DE DADES	11
Il·lustració 15: DIAGRAMA DE GANTT	12

14 ANNEXOS

Il·lustració 13: DIAGRAMA DE CASOS



Il·lustració 14: DIAGRAMA DE BASE DE DADES



Il·lustració 15: DIAGRAMA DE GANTT

